

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/382503496>

PARKING ALLOTMENT SYSTEM PROJECT REPORT.

Research Proposal · February 2024

DOI: 10.13140/RG.2.2.24889.99683

CITATIONS

0

READS

88

1 author:



Kamal Acharya

Tribhuvan University

249 PUBLICATIONS 4,439 CITATIONS

SEE PROFILE

**AN
INTERNSHIP REPORT
ON
PARKING ALLOTMENT SYSTEM
PROJECT
BY
KAMAL ACHARYA
(Tribhuvan University)**

Date: 2024/02/25

1. INTRODUCTION

1.1 INTRODUCTION

This project is entitled as **“PARKING ALLOTMENT SYSTEM”**, aim of this project it to reduce the traffic in the parking place and reduce the theft. The main objective of this project is to computerize the activities performed in the parking. The front end of this project is VB.Net and the back end is SQL Server.

This project is developed as user friendly software so that it meets the user needs at any time. Information can be created and altered by administrator. The objective of this project is to aid in the optimum processing of data. This is intended to provide the basic for the design and development of a software product that will encompass recording maintenance and reporting of data relating to the various stages of the process.

1.2 ABOUT THE PROJECT

The objective of this project is to aid in the optimum processing of data. This is intended to provide the basic for the design and development of a software product that will encompass recording maintenance and reporting of data relating to the various stages of the process. This project is developed for four square parkings.

This system is windows application software which attempts to integrate all departments and functions across an organization onto a single computer system that can serve all those departments' particular needs. This project is developed as user friendly software so that it meets the user needs at any time.

This project resolves all the existing problems. The database is user friendly, personalizing components are also available in this project. This project maintains separate file for storing parking area information, customer information, employee information, parking information and billing information.

2. SYSTEM STUDY

2.1 INTRODUCTION

As software development is large process so work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. The view of this system is necessary when software must interface with other elements such as hardware, people and other resources.

2.2 EXISTING SYSTEM

In the existing system the process are manually operated systems. In the existing system user can maintain only some details and less billing details. It is difficult to maintain in future. Admin must be taking notes for function details in small note. Thus the note is keep in all times and the note is missing he cannot do anything. So it is very low level of user friendliness.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- The volume of the database is very high
- Manipulating the data requires more storage
- Since, it is not user friendly system
- Data retrieval process to take more time
- Existing System is a tedious process

2.3 PROPOSED SYSTEM

The proposed system will be overcome all of those problems. To develop user friendly software that meets the user needs any time. In proposed system, the Response Time that is accessing the database information provided by the system is good. The Proposed system was accessed to ensure that it reduce the retrieval time and increases the efficiency.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- User friendliness is provided in the application with various controls
- The system makes the overall project management much easier and flexible
- Data is managed with very secure
- There is no risk of data mismanagement at any level while the project development is under process

2.4 SYSTEM SPECIFICATION

2.4.1 HARDWARE SPECIFICATION

Processor	: Intel Core i3
RAM	: 2 GB
Hard disk drive	: 500 GB
Clock Speed	: 2.1 GHz
Monitor	: Colour Monitor
Printer	: Laser Printer
Keyboard	: Multi Media
Mouse	: Optical Mouse

2.4.2 SOFTWARE SPECIFICATION

Technology	: Microsoft Visual Studio 2010
Front end	: VB.Net 2010
Back end	: SQL Server 2008

ABOUT THE SOFTWARE

VB .NET 2010

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate.

There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

- To provide a consistent object-oriented programming environment whether object Codes is stored and executed locally on Internet-distributed, or executed remotely.
- To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
- Eliminates the performance problems.

The common language runtime is the foundation of the .NET Framework. It manages code at execution time, providing important services such as memory management, thread management, monitoring and also ensures more security and robustness.

The concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

It is a comprehensive, object-oriented collection of reusable types used to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime to enables embeds managed components or Windows Forms controls in HTML documents.

The common language runtime manages memory; thread execution, code execution, code safety verification, compilation, and other system services these are all run on CLR.

- Security
- Robustness
- Productivity

➤ Performance

The runtime enforces code access security. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers.

The runtime is designed to enhance performance. The common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language.

Microsoft .NET supports not only language independence, but also language integration. This means that it can inherit from classes, catch exceptions, and take advantage of polymorphism across different languages. The .NET Framework makes this possible with a specification called the Common Type System (CTS) that all .NET components must obey.

For example, everything in .NET is an object of a specific class that derives from the root class called System. Object. The CTS supports the general concept of classes, interfaces, delegates (which support call-backs), reference types, and value types.

Garbage Collection

Garbage Collection is another new feature in Visual Basic.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use. In Visual Basic.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

Overloading

Overloading is another feature in Visual Basic.NET. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of

arguments. Besides using overloading for procedures, and use it for constructors and properties in a class.

Multithreading

Visual Basic.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, it can use multithreading to decrease the time taken by an application to respond to user interaction. To decrease the time taken by an application to respond to user interaction, it must ensure that a separate thread in the application handles user interaction.

Structured Exception Handling

Visual Basic.NET supports structured handling, which enables to detect and remove errors at runtime. In Visual Basic.NET, need to use Try...Catch...Finally statements to create exception handlers. Using Try...Catch...Finally statements, it can create robust and effective exception handlers to improve the performance of the application.

SQL SERVER 2008

SQL Server adds certain features including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services.

SQL Server new features and enhancements include Always On SQL Server Failover Cluster Instances and Availability Groups which provides a set of options to improve database availability Contained Databases which simplify the moving of databases between instances, new and modified Dynamic Management Views and Functions, programmability enhancements including new spatial features, metadata discovery, sequence objects and the THROW statement, performance enhancements such as Column Store Indexes as well as improvements to On Line and partition level operations and security enhancements including provisioning during setup, new permissions, improved role management, and default schema assignment for groups.

3. SYSTEM DESIGN

3.1 INTRODUCTION

System design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture. The system design includes following design process.

3.2 INPUT DESIGN

The input design comprises the forms through which the data can be entered. A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy. The form view displays the whole design of the form.

To build or modify the structure of a form, we work in forms design view. We can add control to the form that are bound to fields in a table or query, includes textboxes, option buttons, grid view and pictures.

There should be away of changing the initial state of the problem. This is most usually a person's knowledge or skill level. For instance, a computer programmer presented with a problem would utilize his or her knowledge of programming languages to transform the state of the problem. This can be rectified easily by the input design.

The input forms in these projects are,

- Customer Form
- Employee Form
- Parking Area Form
- Vehicle Inlet Form
- Vehicle Outlet Form
- Billing Form

3.3 TABLE DESIGN

1) **Table Name:** Customer Details

Primary Key: Cus_ID

Field Name	Data Type	Size	Description	Constraints
Cus_ID	nvarchar	10	Customer Identification Number	Primary Key
Cus_name	nvarchar	25	Name of the Customer	Not Null
Cus_cno	bigint	10	Contact number of the Customer	Not Null
Cus_emid	bigint	30	Email ID of the Customer	Not Null
Cus_addr	nvarchar	50	Address of the Customer	Not Null
Rdate	Date	—	Date of Registration	Not Null

2) Table Name: Employee Details

Primary Key: Emp_ID

Field Name	Data Type	Size	Description	Constraints
Emp_ID	nvarchar	10	Employee Identification Number	Primary Key
Emp_name	nvarchar	25	Name of Employee	Not Null
Emp_cno	bigint	10	Contact Number of Employee	Not Null
Emp_Addr	nvarchar	50	Address of Employee	Not Null
DOJ	Date	—	Date of Join	Not Null
Sal	float	5	Salary	Not Null

3) Table Name: Parking Area Details

Primary Key: PA_ID

Field Name	Data Type	Size	Description	Constraints
PA_ID	bigint	4	Parking Area Identification Number	Primary Key
P_Type	nvarchar	25	Parking Type	Not Null
PA_Type	number	10	Parking Area Type	Not Null
PA_Rpd	float	5	Parking Area Rent per day	Not Null

4) Table Name: Vehicle Inlet Details

Primary Key: VI_ID

Foreign Key: Cus_ID, PA_ID

Field Name	Data Type	Size	Description	Constraints
VI_ID	bigint	7	Vehicle Inlet Identification Number	Primary Key
Cus_ID	nvarchar	10	Customer Identification Number	Foreign Key
Cus_name	nvarchar	25	Name of the Customer	Not Null
V_Typ	nvarchar	14	Vehicle Type	Not Null
V_Nam	nvarchar	50	Vehicle Name	Not Null
V_Rno	nvarchar	14	Vehicle Register Number	Not Null
PA_ID	bigint	4	Parking Area Identification Number	Foreign Key
P_Type	nvarchar	25	Parking Type	Not Null
PA_Type	number	10	Parking Area Type	Not Null
PA_Rpd	float	5	Parking Area Rent per day	Not Null
VI_Dat	date	—	Vehicle Inlet Date	Not Null
VI_Tim	time	—	Vehicle Inlet Time	Not Null

5) Table Name: Vehicle Outlet Details

Primary Key: VO_ID

Foreign Key: VI_ID, Cus_ID, PA_ID

Field Name	Data Type	Size	Description	Constraints
VO_ID	bigint	7	Vehicle Outlet Identification Number	Primary Key
VI_ID	bigint	7	Vehicle Inlet Identification Number	Foreign Key
Cus_ID	nvarchar	10	Customer Identification Number	Foreign Key
Cus_name	nvarchar	25	Name of the Customer	Not Null
V_Typ	nvarchar	14	Vehicle Type	Not Null
V_Nam	nvarchar	50	Vehicle Name	Not Null
V_Rno	nvarchar	14	Vehicle Register Number	Not Null
PA_ID	bigint	7	Parking Area Identification Number	Foreign Key
P_Type	nvarchar	25	Parking Type	Not Null
PA_Type	number	10	Parking Area Type	Not Null
PA_Rpd	float	5	Parking Area Rent per day	Not Null
VO_Dat	date	—	Vehicle Outlet Date	Not Null
VO_Tim	time	—	Vehicle Outlet Time	Not Null

6) Table Name: Billing Details

Primary Key: B_ID

Foreign Key: VO_ID, Cus_ID, PA_ID

Field Name	Data Type	Size	Description	Constraints
B_ID	bigint	7	Billing Identification Number	Primary Key
VO_ID	bigint	7	Vehicle Outlet Identification Number	Foreign Key
Cus_ID	nvarchar	10	Customer Identification Number	Foreign Key
Cus_name	nvarchar	25	Name of the Customer	Not Null
V_Typ	nvarchar	14	Vehicle Type	Not Null
V_Nam	nvarchar	50	Vehicle Name	Not Null
V_Rno	nvarchar	14	Vehicle Register Number	Not Null
PA_ID	bigint	7	Parking Area Identification Number	Foreign Key
P_Type	nvarchar	25	Parking Type	Not Null
PA_Type	number	10	Parking Area Type	Not Null
PA_Rpd	float	5	Parking Area Rent per day	Not Null
No_Pa	float	5	Number of days Parked	Not Null
B_Dat	date	—	Billing Date	Not Null
B_Tim	time	—	Billingt Time	Not Null

3.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system, the output design determines the input to be given to the application. The output is designed in such a way that it is attractive, convenient and informative. Forms are designed in VB.NET with various features, which make the console output more pleasing. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information.

The output forms in these projects are,

- Customer Report
- Parking Area Report
- Vehicle Inlet Report
- Vehicle Outlet Report
- Billing Report

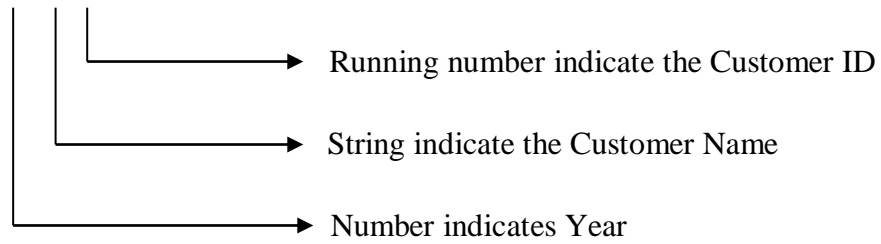
3.5 CODE DESIGN

The code in the system enables easy way of updating and identifying the items easily. It normal practice that all the primary keys should have a codification scheme based on the codification scheme the data can be allowed to enter. This increases the flexibility as user cannot give any type of data which will affect the system performance.

A design code is a document that sets rules for the design of a new development in the system design. It is a tool that can be used in the design and planning process, but goes further and is more regulatory than other forms of guidance commonly used in the English planning system over recent decades. It can be thought of as a process and document – and therefore a mechanism – which operationalises design guidelines or standards which have been established through a masterplan process. The masterplan or design framework is the vision. It should be accompanied by a design rationale that explains the objectives, with the design code providing instructions to the appropriate degree or precision of the more detailed design work.

Table Name: Customer Details

99XXX99



Example

19CUS01

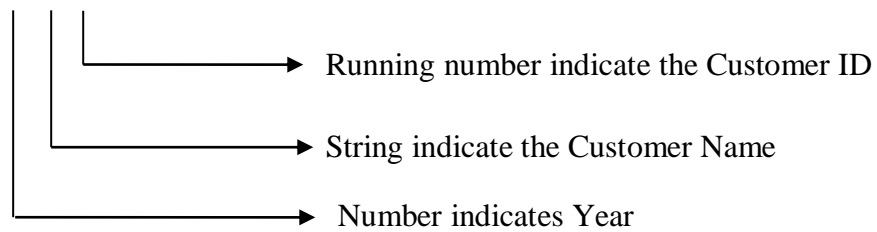
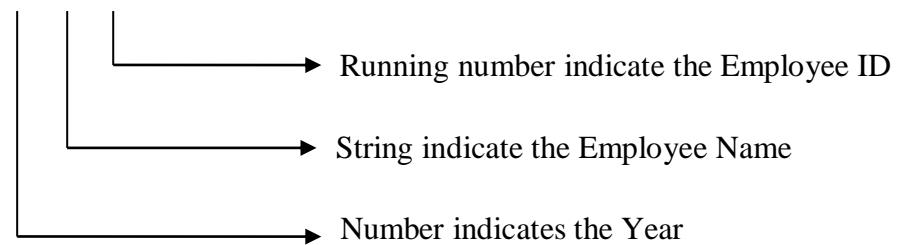


Table Name: Employee Details

99XXX99



Example

19EMP01

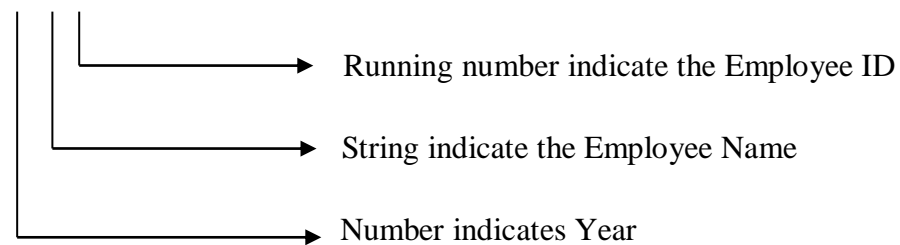
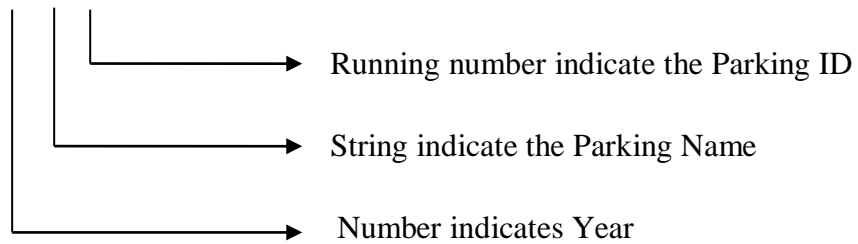


Table Name: Parking Details

99XXX99



Example

19PAR01

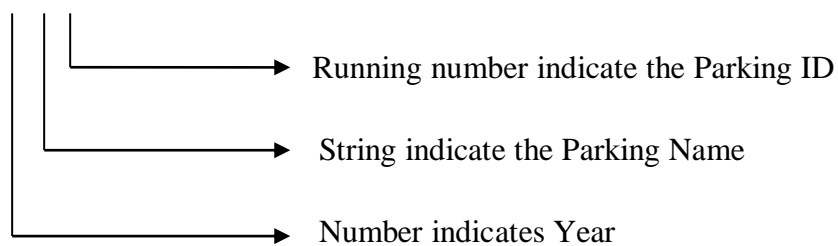
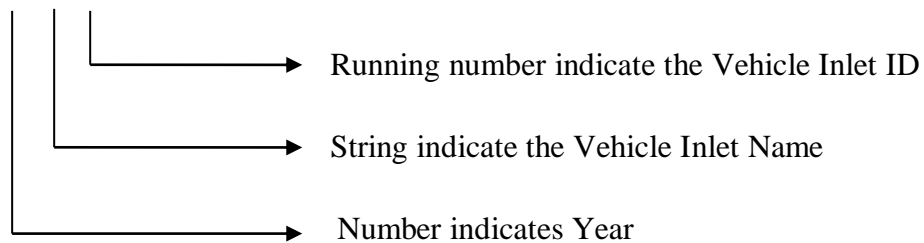


Table Name: Vehicle Inlet Details

99XXX99



Example

19VII01

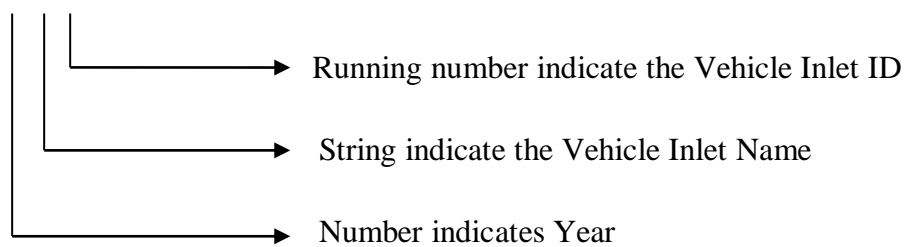
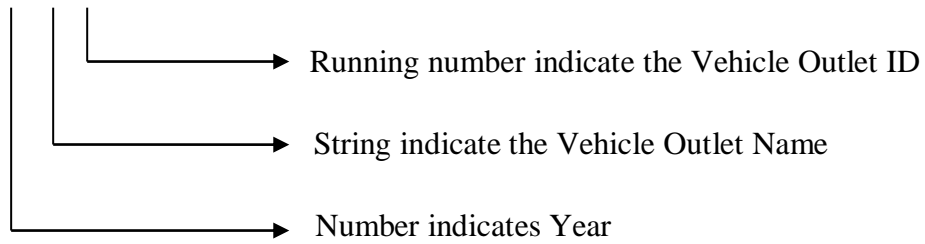


Table Name: Vehicle Outlet Details

99XXX 99



Example

19VOI01

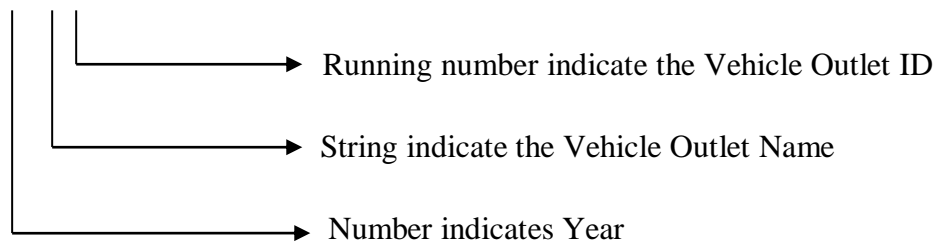
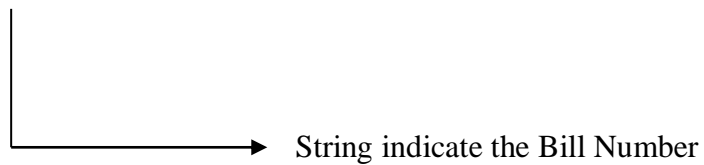


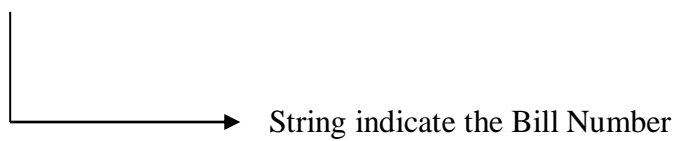
Table Name: Billing Details

999999



Example

000001



4. SYSTEM DEVELOPMENT AND TESTING

4.1 INTRODUCTION

Testing is the process of confirming that a program or system does what it is proposed off, Testing is the only way to assure the quality of software and it is an umbrella activity rather than a separate phase. This is an activity to be performed in parallel with the software efforts and one that consists of its own phase of analysis, design, implementation, execution and maintenance.

During analysis and design, a software verification plan and acceptance test plan is prepared. The verification plan describes the methods to be used in verifying that the requirements are satisfied by the design documents and that the source is consistent with the requirements specification and design documents. The acceptance test plan includes test cases, outcomes and capabilities demonstrated by each test case. Following completion of the verification plan and Acceptance plan, a software verification review is held to evaluate the adequacy of the plans.

4.2 SYSTEM DEVELOPMENT

During product evolution, in-process audits are conducted to verify consistency and completeness of the work products. Items to be audited for consistency include interface specification for hardware and software: internal design versus functional requirements versus test descriptions. Implementation is the phase where the system goes for actual functioning. Hence in this phase one has to be cautious because all the efforts undertaken during the project will be fruitful only if the software is properly implemented according to the plans made.

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation and file conversion. Depending on the nature of the system, extensive user training may be required. The initial parameters of the development should be modified as the result of programming efforts; programming provides a reality test for the assumptions made by the analyst.

4.3 SYSTEM TESTING

Unit Testing

This testing method considers a module as single unit and checks the unit at interfaces and communities with other modules rather than getting into details at statement level. Here the module will be treated as BLACKBOX, which will take some inputs and generate output. Outputs for a given set of input combination are pre calculated and are generated by the module.

Integration Testing

Here all the pre-tested individual modules will be assembled to create a larger system and tests are carried out at system level to make sure that all modules are working with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually and are also running cohesion with other modules. For this testing we create test-cases to check all modules once and then a generated test combination of test paths with the system to make sure that no path is making its way into chaos.

Validation Testing

Testing is major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. To uncover errors that are associated with interfacing the objective is to make test modules and built a program structure that has detected by design. In a non-incremental integration all the modules are combined in advance and the program is tested as a whole. Here error will appear in an end-less loop function. In validation testing the program is constructed and tested in small segments where the errors are isolated and corrected.

5. CONCLUSION

As per a survey, most consumers are impulsive and usually make a decision to stay on an application within the first few seconds. If the parking looks poor or like hundreds of others, the customer is most likely to skip to the other site. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible. It eased the user with the less typing of messages and the user can select everything in simply selecting the options available under menu. Since the System is highly modularized, each part of the system can be implemented whenever necessary.

The goal of this project is to develop the system that was found to be accurate and user friendly. So the system was treated with all the sample data. The system also provides full security and high reliability. The security helps the user by not allowing any unauthorized person to enter the system and access the personal data. As focused this system has a great facility to all users. All users can benefit using this project. We hope that our project may satisfy the user. The up gradation if any to the system can be done without affecting proper functioning of the system. The documentation provided helps a person with a minimum amount of knowledge to understand it well.

FUTURE ENHANCEMENT

This application Also, if you see at the first glance that you find it to be complete but we want to make it still mature and fully automatic. As system is flexible you can generate more report and screen as and when required. The system is modified in future as per the owner requirement. In this system we can add more reports.

References

- Acharya, Kamal. "STUDENT INFORMATION MANAGEMENT SYSTEM." *Authorea Preprints* (2023).
- Acharya, Kamal. "Library Management System." Available at SSRN4807104 (2019).
- ACHARYA, KAMAL, et al. "LIBRARY MANAGEMENT SYSTEM." (2019).
- Acharya, Kamal. "Online bus reservation system project report." *Authorea Preprints* (2024).
- Acharya, Kamal. "Online bus reservation system project report." (2024).
- Acharya, Kamal. "Online Bus Reservation System." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "Student Information Management System Project." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal. "ATTENDANCE MANAGEMENT SYSTEM." *International Research Journal of Modernization in Engineering Technology and Science* (2023): n. pag.
- Acharya, Kamal. "College Information Management System." *SSRN ElectroNIC ASIA Journal* (2024): n. pag.
- Acharya, Kamal, Attendance Management System Project (April 28, 2024). Available at SSRN: <https://ssrn.com/abstract=4810251> or <http://dx.doi.org/10.2139/ssrn.4810251>
- Acharya, Kamal, Online Food Order System (May 2, 2024). Available at SSRN: <https://ssrn.com/abstract=4814732> or <http://dx.doi.org/10.2139/ssrn.4814732>
- Acharya, Kamal, University management system project. (May 1, 2024). Available at SSRN: <https://ssrn.com/abstract=4814103> or <http://dx.doi.org/10.2139/ssrn.4814103>
- Acharya, Kamal, Online banking management system. (May 1, 2024). Available at SSRN: <https://ssrn.com/abstract=4813597> or <http://dx.doi.org/10.2139/ssrn.4813597>
- Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: <https://ssrn.com/abstract=4817534> or <http://dx.doi.org/10.2139/ssrn.4817534>
- Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819626> or <http://dx.doi.org/10.2139/ssrn.4819626>
- Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819630> or <http://dx.doi.org/10.2139/ssrn.4819630>
- Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831694> or <http://dx.doi.org/10.2139/ssrn.4831694>

Acharya, Kamal, Online job placement system project report. (January 10, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4831638> or <http://dx.doi.org/10.2139/ssrn.4831638>

Acharya, Kamal, Software testing for project report. (May 16, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4831028> or <http://dx.doi.org/10.2139/ssrn.4831028>

Acharya, Kamal, ONLINE CRIME REPORTING SYSTEM PROJECT. (August 10, 2022). Available at
 SSRN: <https://ssrn.com/abstract=4831015> or <http://dx.doi.org/10.2139/ssrn.4831015>

Acharya, Kamal, Burger ordering system project report. (October 10, 2022). Available at
 SSRN: <https://ssrn.com/abstract=4832704> or <http://dx.doi.org/10.2139/ssrn.4832704>

Acharya, Kamal, Teachers Record Management System Project Report (December 10, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4833821> or <http://dx.doi.org/10.2139/ssrn.4833821>

Acharya, Kamal, Dairy Management System Project Report (December 20, 2020). Available at SSRN: <https://ssrn.com/abstract=4835231> or <http://dx.doi.org/10.2139/ssrn.4835231>

Acharya, Kamal, Electrical Shop Management System Project (December 10, 2019). Available at
 SSRN: <https://ssrn.com/abstract=4835238> or <http://dx.doi.org/10.2139/ssrn.4835238>

Acharya, Kamal, Online book store management system project report. (February 10, 2020). Available at
 SSRN: <https://ssrn.com/abstract=4835277> or <http://dx.doi.org/10.2139/ssrn.4835277>

Acharya, Kamal, Paint shop management system project report. (January 10, 2019). Available at
 SSRN: <https://ssrn.com/abstract=4835441> or <http://dx.doi.org/10.2139/ssrn.4835441>

Acharya, Kamal, Supermarket billing system project report. (August 10, 2021). Available at SSRN: <https://ssrn.com/abstract=4835474> or <http://dx.doi.org/10.2139/ssrn.4835474>

Acharya, Kamal, Online taxi booking system project report. (March 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4837729> or <http://dx.doi.org/10.2139/ssrn.4837729>

Acharya, Kamal, Online car servicing system project report. (March 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4837832> or <http://dx.doi.org/10.2139/ssrn.4837832>

Acharya, Kamal, School management system project report. (July 10, 2021). Available at SSRN: <https://ssrn.com/abstract=4837837> or <http://dx.doi.org/10.2139/ssrn.4837837>

Acharya, Kamal, Furniture Showroom Management System Project Report (March 21, 2021). Available at
 SSRN: <https://ssrn.com/abstract=4839422> or <http://dx.doi.org/10.2139/ssrn.4839422>

Acharya, Kamal, Online Vehicle Rental System Project Report (March 21, 2019). Available at SSRN: <https://ssrn.com/abstract=4839429> or <http://dx.doi.org/10.2139/ssrn.4839429>

Acharya, Kamal, Fruit Shop Management System Project Report (August 10, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4841048> or <http://dx.doi.org/10.2139/ssrn.4841048>

Acharya, Kamal, Hall Booking Management System Project Report (December 21, 2023). Available at
 SSRN: <https://ssrn.com/abstract=4841055> or <http://dx.doi.org/10.2139/ssrn.4841055>

Acharya, Kamal, *Lundry Management System Project Report* (October 21, 2023). Available at SSRN: <https://ssrn.com/abstract=4841059> or <http://dx.doi.org/10.2139/ssrn.4841059>

Acharya, Kamal, *A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT* (September 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4841209> or <http://dx.doi.org/10.2139/ssrn.4841209>

Acharya, Kamal, *A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT* (May 25, 2024). Available at SSRN: <https://ssrn.com/abstract=4841210> or <http://dx.doi.org/10.2139/ssrn.4841210>

Acharya, Kamal, *ONLINE DATING MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4842066> or <http://dx.doi.org/10.2139/ssrn.4842066>

Acharya, Kamal, *ONLINE RESUME BUILDER MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2021). Available at SSRN: <https://ssrn.com/abstract=4842071> or <http://dx.doi.org/10.2139/ssrn.4842071>

Acharya, Kamal, *TOLL TEX MANAGEMENT SYSTEM PROJECT REPORT* (August 21, 2023). Available at SSRN: <https://ssrn.com/abstract=4842082> or <http://dx.doi.org/10.2139/ssrn.4842082>

Acharya, Kamal, *Chat Application Through Client Server Management System Project Report* (June 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4842761> or <http://dx.doi.org/10.2139/ssrn.4842761>

Acharya, Kamal, *Web Chatting Application Management System Project Report* (April 25, 2022). Available at SSRN: <https://ssrn.com/abstract=4842771> or <http://dx.doi.org/10.2139/ssrn.4842771>

Acharya, Kamal, *Automobile management system project report* (May 25, 2022). Available at SSRN: <https://ssrn.com/abstract=4846917> or <http://dx.doi.org/10.2139/ssrn.4846917>

Acharya, Kamal, *College bus management system project report* (April 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4846920> or <http://dx.doi.org/10.2139/ssrn.4846920>

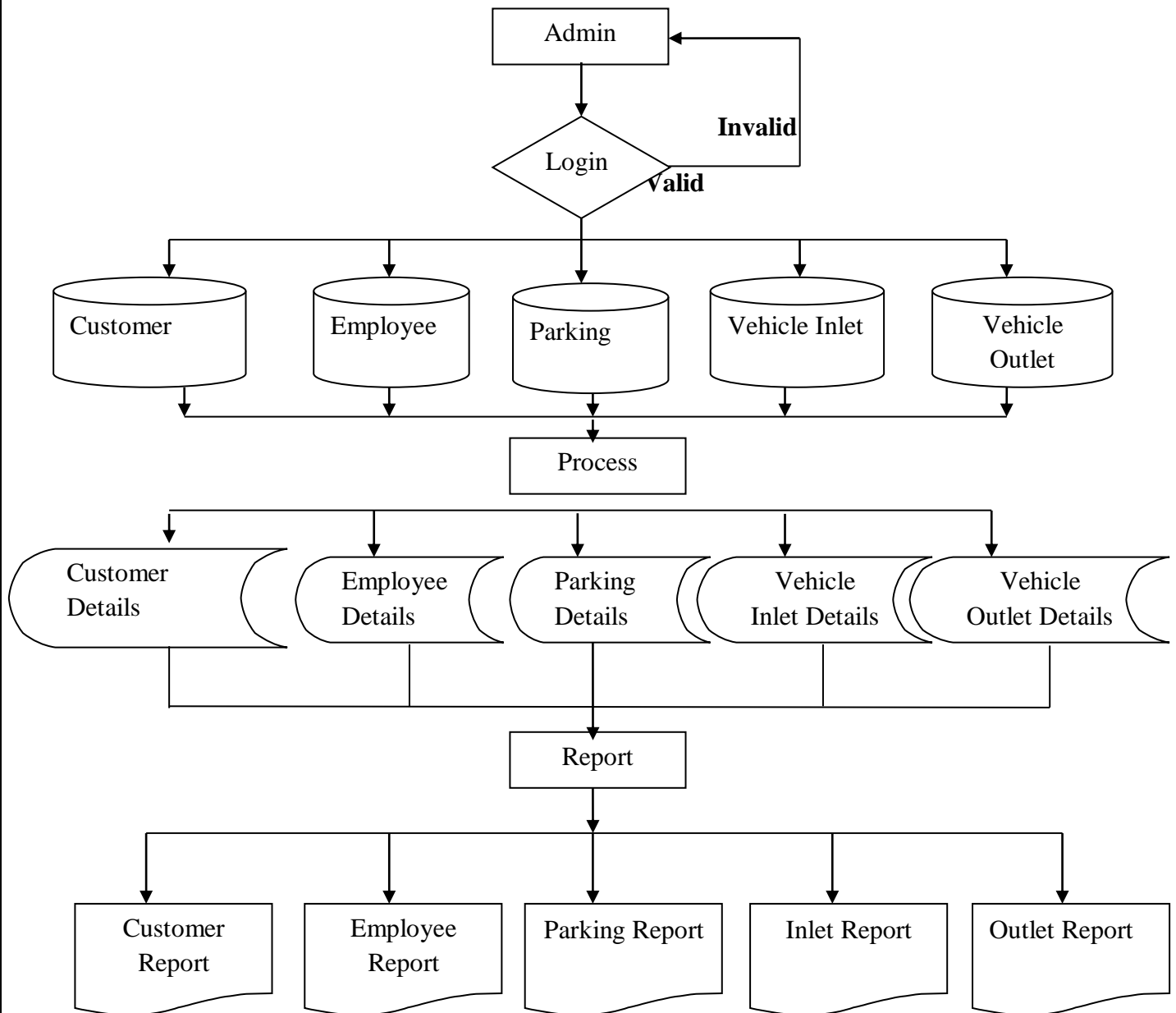
Acharya, Kamal, *Courier management system project report* (May 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4846922> or <http://dx.doi.org/10.2139/ssrn.4846922>

Acharya, Kamal, *Event management system project report* (April 25, 2021). Available at SSRN: <https://ssrn.com/abstract=4846927> or <http://dx.doi.org/10.2139/ssrn.4846927>

Acharya, Kamal, *Library management system project report II* (May 25, 2020). Available at SSRN: <https://ssrn.com/abstract=4848857> or <http://dx.doi.org/10.2139/ssrn.4848857>

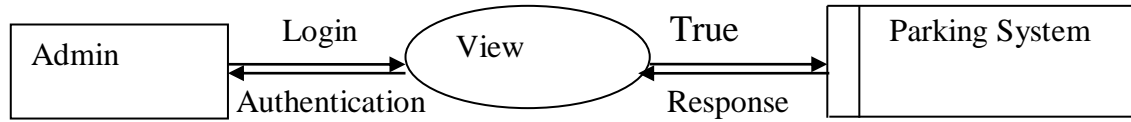
APPENDICES

APPENDIX A: SYSTEM FLOW DIAGRAM

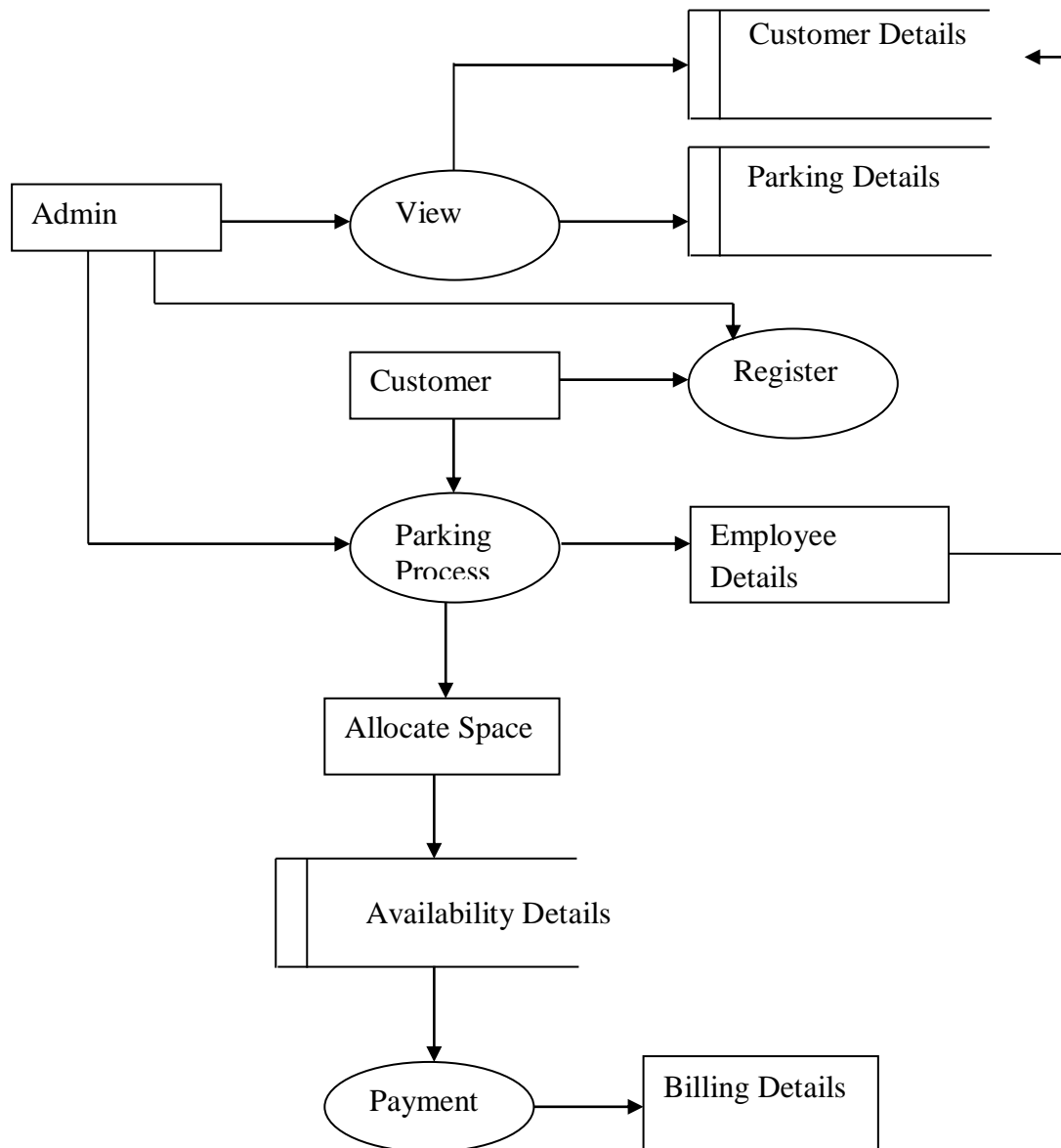


APPENDIX B: DATA FLOW DIAGRAM

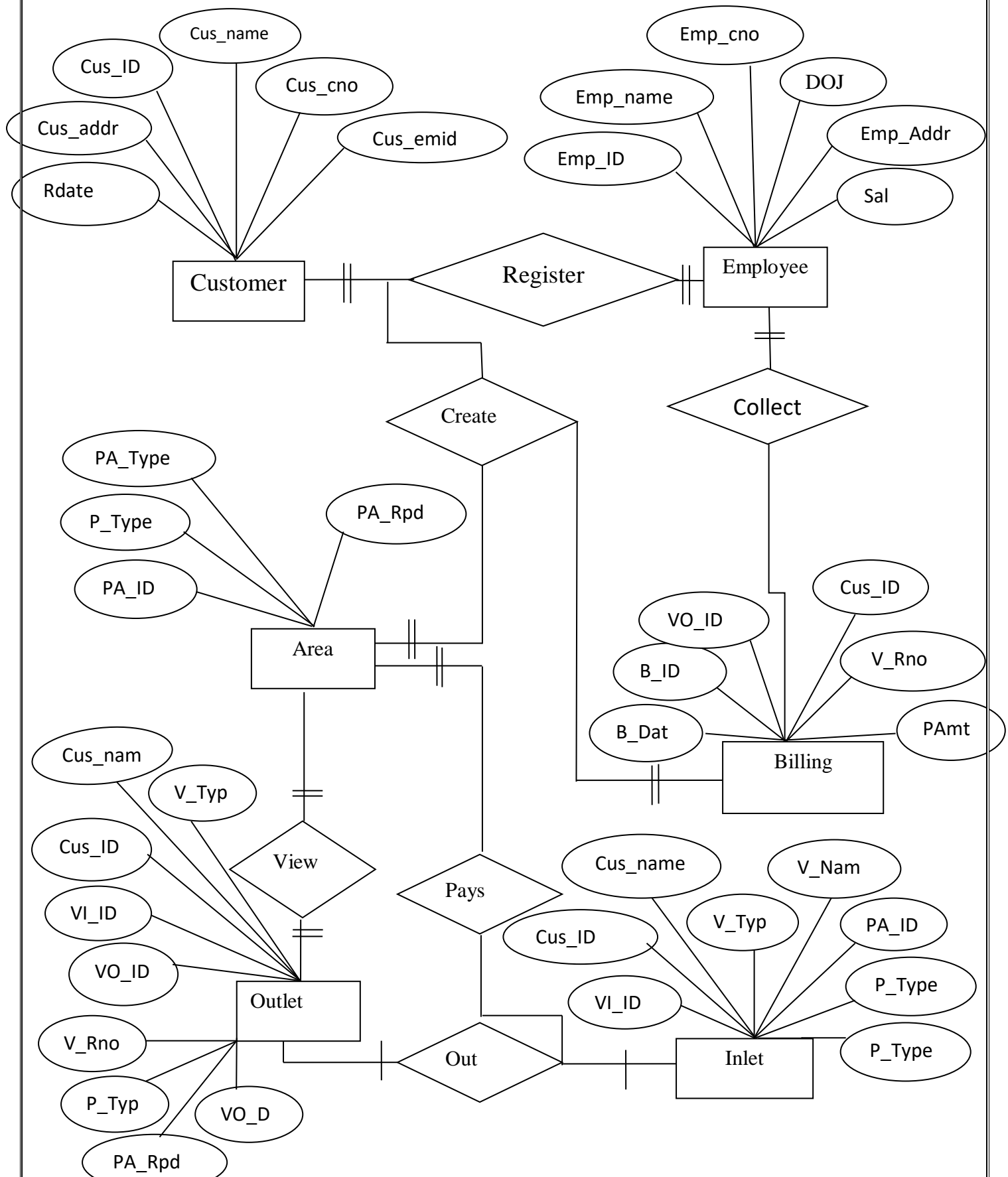
Level 0



Level 1



APPENDIX C: ENTITY RELATIONSHIP DIAGRAM



APPENDIX D: SAMPLE SCREENS

APPENDIX E: SAMPLE REPORT

SAMPLE CODING:

```
Imports System.Data
Imports System.Data.SqlClient
Public Class Customer
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim da As New SqlDataAdapter("select *from tblCustomer", con)
    Dim ds As New DataSet
    Dim dbind As New BindingSource
    Dim a As Integer = 0
    Dim qry, qry1 As String
    Dim i As Integer
    Dim reader As SqlDataReader
    Dim adp As SqlDataAdapter
    Private Sub Customer_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
        LoadDatabase()
        Load1()
    End Sub
    Private Sub LoadDatabase()
        con.Close()
        con.ConnectionString = "Data Source=.\SQLEXPRESS;Initial Catalog=Parking;Integrated
Security=True"
        con.Open()
    End Sub
    Public Sub Load1()
        ClearGridView()
        LoadDatabase()
        da.Fill(ds, "tblCustomer")
        dbind.DataSource = ds
        dbind.DataMember = ds.Tables(0).ToString()
        DataGridView1.DataSource = dbind
    End Sub
    Private Sub ClearGridView()
        For Each row As DataGridViewRow In DataGridView1.Rows
            DataGridView1.Rows.Remove(row)
        Next
    End Sub
    Public Sub clear()
        txtEmpid.Clear()
        txtEmpname.Clear()
        TextBox5.Clear()
        TextBox7.Clear()
        TextBox8.Clear()
    End Sub
    Private Sub DataGridView1_SelectionChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles DataGridView1.SelectionChanged
        Try
            Dim dgv As DataGridView = TryCast(sender, DataGridView)
            If dgv IsNot Nothing AndAlso dgv.SelectedRows.Count > 0 Then
                Dim row As DataGridViewRow = dgv.SelectedRows(0)
                If row IsNot Nothing Then
                    txtEmpid.Text = row.Cells(0).Value.ToString()
                End If
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub
End Class
```

```

        txtEmpname.Text = row.Cells(1).Value.ToString()
        TextBox5.Text = row.Cells(2).Value.ToString()
        TextBox8.Text = row.Cells(3).Value.ToString()
        TextBox7.Text = row.Cells(4).Value.ToString()
        dt DOJ.Value = row.Cells(5).Value.ToString()
    End If
End If
Catch ex As Exception
End Try
End Sub

```

```

Private Sub btInsert_Click(sender As System.Object, e As System.EventArgs) Handles
btInsert.Click
    LoadDatabase()
    cmd.Parameters.Clear()
    Try
        cmd.Connection = con
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "spInsertCust"
        cmd.Parameters.AddWithValue("@CID", txtEmpid.Text)
        cmd.Parameters.AddWithValue("@CName", txtEmpname.Text)
        cmd.Parameters.AddWithValue("@CNo", TextBox5.Text)
        cmd.Parameters.AddWithValue("@EM", TextBox8.Text)
        cmd.Parameters.AddWithValue("@Addr", TextBox7.Text)
        cmd.Parameters.AddWithValue("@DOR", dt DOJ.Value)
        cmd.ExecuteNonQuery()
        MsgBox("Details of Customer are Successfully Inserted", vbInformation, "Successfully
Inserted")
        clear()
        con.Close()
    Catch ex As Exception
        MsgBox("Details are not Inserted! Please Try Later", vbInformation, "Error In Inserting
Company Details")
    End Try
    ClearGridView()
    ClearGridView()
    ClearGridView()
    ClearGridView()
    ClearGridView()
    ClearGridView()
    ClearGridView()
    ClearGridView()

    Load1()
End Sub

```

```

Private Sub btUpdate_Click(sender As System.Object, e As System.EventArgs) Handles
btUpdate.Click
    LoadDatabase()
    Me.Refresh()
    'Dim answer As Long
    Try
        cmd.Parameters.Clear()
        'answer = MsgBox("Are you sure to Updating Data?", vbYesNo, "Updating Record")
        'If (answer = vbYes) Then
        cmd.Connection = con

```

```

cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = "spUpdateCust"
'cmd.Parameters.Add("@ProductCode", SqlDbType.NVarChar).Value = ttProductCode.Text
cmd.Parameters.AddWithValue("@CName", txtEmpname.Text)
cmd.Parameters.AddWithValue("@CNo", TextBox5.Text)
cmd.Parameters.AddWithValue("@EM", TextBox8.Text)
cmd.Parameters.AddWithValue("@Addr", TextBox7.Text)
cmd.Parameters.AddWithValue("@DOR", dt doj.Value)

cmd.Parameters.Add("@CID", SqlDbType.NVarChar).Value = txtEmpid.Text
cmd.ExecuteNonQuery()
MsgBox("Customer Details Updated", vbInformation, "Succesfully Updated")
clear()
con.Close()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
Load1()
Me.Refresh()
'Else
'txtSerialNo.Focus()
"End If
Catch ex As Exception
MsgBox("Unable to Update Record! Please Try Later", vbInformation, "Error in Updating
Record")
End Try
End Sub

Private Sub btDelete_Click(sender As System.Object, e As System.EventArgs) Handles
btDelete.Click
LoadDatabase()
Me.Refresh()
'Dim answer As Long
Try
cmd.Parameters.Clear()
'answer = MsgBox("Are you sure to Deleting Data?", vbYesNo, "Delete Record")
'If (answer = vbYes) Then
cmd.Connection = con
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = "spDeleteCust"
cmd.Parameters.Add("@CID", SqlDbType.NVarChar).Value = txtEmpid.Text
'cmd.Parameters.AddWithValue("@ProductCode", ttProductCode.Text)
cmd.ExecuteNonQuery()
MsgBox("Details Deleted", vbInformation, "Successfully Deleted")
clear()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
ClearGridView()
Load1()

```



```

        Me.Refresh()
    'Else
    '    txtSerialNo.Focus()
    'End If
    Catch ex As Exception
        MsgBox("Unable to Delete Record! Please Try Later", vbInformation, "Error in Deleting")
    End Try
End Sub

Private Sub btClear_Click(sender As System.Object, e As System.EventArgs) Handles
btClear.Click
    txtEmpid.Clear()
    txtEmpname.Clear()
    TextBox5.Clear()
    TextBox7.Clear()
    TextBox8.Clear()

End Sub

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles
Button1.Click
    MDIMenu.Show()
    Me.Hide()

End Sub

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles
Button2.Click
    End

End Sub

Private Sub txtSearch_TextChanged(sender As System.Object, e As System.EventArgs) Handles
txtSearch.TextChanged
    Dim con As New SqlConnection
    con = New SqlConnection()
    con.ConnectionString = "Data Source=.\SQLEXPRESS;Initial Catalog=Parking;Integrated
Security=True"
    con.Open()
    Dim da As New SqlDataAdapter("spSearchCust", con)
    da.SelectCommand.Parameters.Clear()
    da.SelectCommand.CommandType = CommandType.StoredProcedure
    da.SelectCommand.Parameters.AddWithValue("CName", txtSearch.Text)
    Dim dt As New DataTable
    da.Fill(dt)
    DataGridView1.DataSource = dt
    con.Close()
End Sub
End Class

```